COORDINATED HIGHWAYS ACTION RESPONSE TEAM
STATE HIGHWAY ADMINISTRATION

WO47 CHART Web Map Arch Refresh

Detailed Design

Contract SHA-06-CHART
Document # WO47-DS-001
Work Order 42, Deliverable 02

November 24, 2014
By
CSC

Revision	Description	Pages Affected	Date
0	Initial Release	All	11/24/2014

Table of Contents

Table of Figures

1 Introduction

1.1 Purpose

This document describes the high level design of the software for WO47 CHARTWeb Map Architecture Refresh.

Following is a summary of the major features provided by WO47:

1.1 Web Map Architecture Refresh

This release will update the existing CHARTWeb Google Map to apply the recommendations provided by the WO08 TR92 CHARTWeb Map Architecture Review. These recommendations include:

- Add a caching mechanism to avoid going to the database for each request from the map.
- Add classes to hold data for each layer on the map (Camera, CCTV, DMS, Event, EORS Closure, EORS Restriction, External Camera, HAR, IPS, RWIS, SEP, and TSS).
- Rewrite handlers that provide data for the map to update the objects in the cache instead of returning XML to the map. One handler for each layer (Camera, CCTV, DMS, Event, EORS Closure, EORS Restriction, External Camera, HAR, IPS, RWIS, SEP, and TSS).
- Add code to return JSON to the map (instead ofXML) based on the objects in the cache (using the extent of the viewport in the map).
- Convert the map from using the Google API to using OpenLayers API.
- Add clustering support to the map.
- Update the Ajax support
- Add JavaScript code for each of the map layers (Camera, CCTV, DMS, Event, EORS Closure, EORS Restriction, External Camera, HAR, IPS, RWIS, SEP, and TSS).
- Provide a map user interface that utilizes Response Web Design.

Assumptions: The following assumptions are made regarding this set of recommendations

1. A re-design of the user interface for the map on the CHART website is not required or requested. Interface changes can be recommended where they make sense with the other changes being recommended.

1.2 CHARTWeb Application Updates

The WO 47 CHARTWeb Map Architecture Refresh will also provide several non-mapping application updates from documented PRs. Those PRs include the following:

PR7139 - Remove CHART Software Development Program Documentation section from CW Reading Room
PR7119 - Investigate the need to modify the CHARTWeb incident data feed
PR7103 - XML Incident feeds are not providing lat/long data

PR6667 - Maryland Weather Station Data sort column headers redirect to data definitions page.
PR6825 - Update URL to Automated Permit System
PR7089 - LCP RSS feed not provding complete data set.
PR7053 - Unable to display planned ETL closures
PR7020 - PHP pages on CHARTWeb sometimes display an Access Violation error
PR6983 - Deprecate use of WT_EORS_CLOSURE table instead use LCP_VW_CHARTWEB_Closures view directly from LCP database
PR6958 - Ham Weather page, Hourly Track page goes into Quirks mode
PR6748 - Investigate potential missing RWIS
PR6637 - IIS Log Files Not Getting Moved To Correct Directory
PR6600 - Event and related traffic alert not displaying on CHART Web
PR6450 - Write special html to the response when an RSS or XML feed has no data

1.2 Objectives

The main objective of this detailed design document is to provide software developers with a framework in which to implement the requirements identified in the CHARTWeb Map Architecture Refresh Requirements document.

1.3 Scope

This design is limited to Phase 3 of the LCP system and associated Mapping application updates. This design does not include designs for components implemented in other releases of the CHARTWeb or CHART systems.

1.4 Design Process

This design is based on a series of Joint Application Design (JAD) sessions that were held with developers, stake holders, and users. The user interface design is included in this document in the Human Machine Interface section. The requirements have been captured as UML Use Case diagrams, also included in this document. The use case diagrams will be the basis for detailed design.

1.5 Design Tools

The use case diagrams, deployment diagram and sequence diagrams will be extracted from the Sparx Enterprise Architect design tool. Within this tool, the design will be contained in the project named “chartweb-WO47-pubmap”.

1.6 Work Products

This design document includes the following work products:

- Architectural Deployment diagram, showing the high level architecture of components related to this project.
- Human Machine Interface, describing the functions of the user interface.
- Use Cases, describing the ways that users interact with the system.
- UML Class diagrams, showing the software objects which allow the system to accommodate the uses of the system described in the Use Case Diagrams.
- UML Sequence diagrams showing how the classes interact to accomplish major functions of the system.

2 Architecture

The sections below discuss specific elements of the architecture and software components that are created, changed, or used in LCP Phase 3.

2.1 Network/Hardware

2.1.1 CHARTWeb

The CHARTWeb Map Architecture Refresh release will require that a new server be added to the MDOT DMZ network.

2.2 Software

2.2.1 CHARTWeb

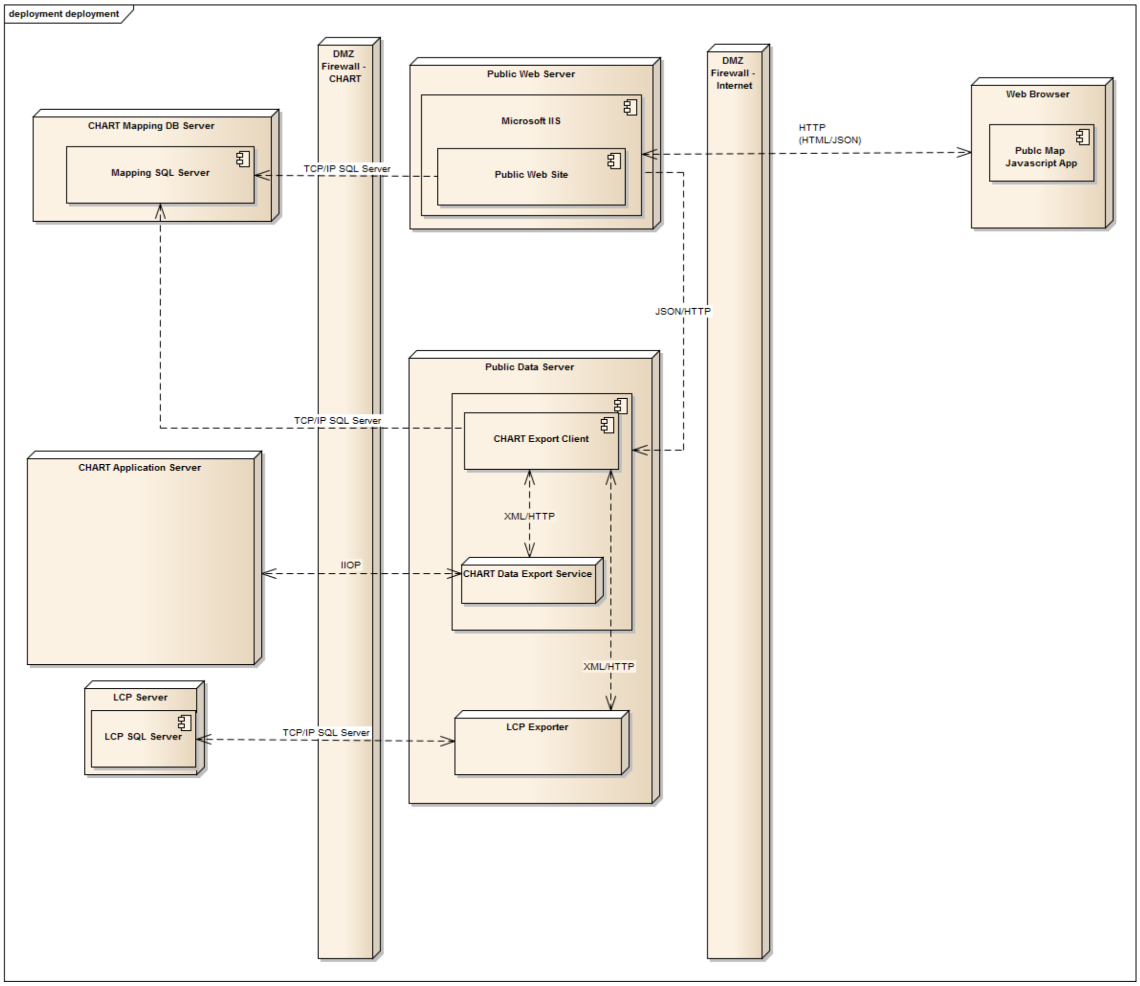
CHARTWeb Map Architecture Refresh will use the Java Servlet development environment that is in use for the CHART Lite GUI and the CHART Web Services. As such it will be able to take advantage of all existing CHART development frameworks that are available.

2.2.2 COTS Products

Product Name	Description
Jackson JSON library for Java	The pubsite package that is being added to the CHART Export Client will use the Jackson library to generate the JSON that will be returned.
Microsoft SQL Server	The CHART Export client will connect to the CHARTWeb SQL Server database using JDBC.
Google Maps	The public web site map will use a Google maps base map.
ExtJS	The public web site map will use the ExtJS JavaScript library.
Openlayers	The Integrated Map feature uses the Open Layers JavaScript API 2.8 (http://openlayers.org/) in order to render interactive maps within a web application without relying on vendor specific software. Open Layers is an open source product released under a BSD style license.
NetTopologySuite	The JTS Topology Suite is an API for modeling and manipulating 2-dimensional linear geometry. It provides numerous geometric predicates and functions. JTS conforms to the Simple Features Specification for SQL published by the Open GIS Consortium.
Log4J	CHART uses the log4J version 1.2.15 for logging purposes
Java Run-Time (JRE)	CHART Export Client uses 1.7.0_45.
JDOM	CHART Export Client uses JDOM b7 (beta-7) dated 2001-07-07. JDOM provides a way to represent an XML

Product Name	Description
	document for easy and efficient reading, manipulation, and writing
JAXB	CHART Export Client uses the jaxb java library to automate the tedious task of hand-coding field-by-field XML translation and validation for exported data.
Apache Tomcat	CHART Export Client uses Apache Tomcat 7.0.47 as the host server

2.2.3 Deployment Interface Compatibility



The above diagram shows the interfaces being added for CHARTWeb with the following details:

- The system will now include a CHART Export Client that is deployed on a “Public Data Server” on the DMZ network. This export client will be added to the Apache Tomcat servlet container that the CHART Data Export Service is already running in. Additionally, an instance of the LCP Exporter will be hosted on this server.
- The LCP Exporter will query the production LCP SQL Server through the firewall that sits between the DMZ and the production network. This will require new ports to be opened through this firewall. This communication is shows as bi-directional because the LCP Exporter must query data from this server and it must also receive SQL Dependency push updates from the SQL Server when the data in the production database tables changes. Note: The team is performing a proof-of-concept activity to determine exactly which ports will have to be opened in the firewall to allow this to happen.
- The CHART Export Client will communicate with the LCP Exporter Web Service via HTTP/XML to retrieve initial inventory and subscribe for permit data updates. It will also

communicate with the CHART Mapping Services deployed on CHART Mapping Application Server via HTTP to notify about permit changes. Permit changes include a change in the geographical location of a permit, new permits added, or any expired/deleted permits.

- The CHART Export Client running on the public data server will get CHART data (DMS, HAR, Traffic Events, TSS, and Cameras) via XML/HTTP communications with the CHART Data Exporter that is running on the same server.
- The CHART Export Client running on the public data server will access EORS V2 data such as Snow Emergency Plans, IPS, and Hauling Restrictions by querying this information from the Mapping SQL Server running in the production environment on the CHART Mapping DB Server. As depicted above queries from the Public Web Server are already made to this database through the firewall. However, if the access to the necessary port is restricted by IP address of the accessing server a new firewall rule may be required.
- All of the data required for the public web site map will be cached in memory within the CHART Export Client running on the public data server. A new module will be added to the CHART Export Client that allows it to return the appropriate data of each type as JSON over HTTP. These requests will be satisfied extremely efficiently because they will access only information that is readily available in memory.
- The Web Browser will access HTML web pages that are hosted on the Microsoft IIS instance running on the Public Web Server as usual. When the user accesses the interactive mapping page on the public web site they will be returned a web page that includes a JavaScript application that sends AJAX requests back to the Microsoft IIS on the Public Web Server to get the data that is needed to display on the map. The Microsoft IIS instance will use a reverse proxy configuration to relay these requests to the CHART Export Client running on the Public Data Server. The CHART Export Client will return the desired JSON; which will then be forwarded back to the requesting client's browser.

2.2.3.1 CHARTWeb Interfaces

2.2.3.1.1 External Interfaces

The map technology refresh adds only JSON/HTTP external interface between the user browser and the Microsoft IIS web server that is described in the section above.

2.2.3.1.2 Internal Interfaces

As described above the CHART Export Client hosted in the DMZ will need to communicate with the following interfaces in order to obtain the data that is required for the mapping application.

- CHART Data Export Service: The CHART Export Client will consume XML data from the CHART Data Export Service in order to keep an up to date cache of data for CHART objects such as Traffic Events, DMS signs, HARs, Detectors and Cameras.
- CHART Web SQL Server database: The CHART Export Client will connect to this database in order to obtain EORS V2 data such as IPS, Snow Emergency Plans and Hauling Restrictions.
- LCP Data Exporter: The CHART Export client will consume XML data from the LCP Data Exporter in order to keep an updated cache of permit data.

2.3 Security

2.3.1 CHARTWeb

CHARTWeb will require no security updates.

2.3.2 CHART Export Client

The CHART Export Client will use the existing digitally signed requests security framework for all communications with the CHART Data Exporter Service. The JSON data feed that the CHART Export Client will expose to the internet for the map application running within the user's browser will not be secured in any way. The JSON is a read only interface and the payload is all publicly available data that will be shown to the user while they are using the map.

2.4 CHARTWeb Data

CHARTweb will be tested with the fielded version of Microsoft SQL Server.

3 Key Design Concepts

3.1 CHARTWeb Map Architecture

- The CHART Export Client is extended to include a cache of data and is deployed in the DMZ. Using the CHART Export Client as a base for the development of the cache allows us to leverage a great deal of code which is already consuming the CHART data that we need in order to populate the CHART Web database. This application is already extended (for another task) to also consume LCP permit data from the LCP exporter. The use of a cache of data elements will make the application far more scalable and far more responsive.
- A LCP Data Exporter is deployed in the DMZ to make permit data available to the CHART Export client that is also running in the DMZ. This allows us to leverage the aforementioned code to consume permits and avoids reliance on the LCP database structure.
- The CHART Web mapping application will utilize the robust ExtJS Javascript framework. This framework provides a complete set of UI components, a robust AJAX infrastructure and a well designed Model View Controller application architecture which will expedite development of the application and provide a very extensible framework that can be easily expanded on in future releases.
- The OpenLayers based mapping application utilizes a two phase request structure when it wants to update the data that it is displaying for the end user. The first request asks for a list of layers that have data that has changed since the last time this client requested data. The response to this request is a JSON array that contains 0 or more layer names that need updating. The mapping application then requests data only for those layers that have been modified. This design will greatly reduce the amount of JSON data that is flowing between the client browser and the application servers and will reduce the CPU load on the client browser by not requiring it to re-draw and re-cluster all objects on all layers once per minute.

3.2 PRs

The PRs for this release are minor enhancements or bug fixes and did not require additional design elements.

3.3 Export Client

The CHART Explort client is extended to add a new module that is capable of returning data in JSON format. This module is implemented to return the data required for all mapping AJAX requests in this release. In future releases it can be expanded to satisfy requests for data from the public site list pages.

Export Client will also communicate with the CHART Mapping Web Services to notify about any new permits, deleted/expired permits, and changes to the geographical locations of the permits.

3.4 Error Processing

3.4.1 Export Client Logging

Export Client will trap all network errors, database errors, xml validation errors and any other internal problems to be logged into a flat file. Enhanced logging to capture the requests/responses exchanged with external web services can also be enabled. The errors reported will require manual inspection when issues are reported.

3.5 Packaging

3.5.1 CHART Map Tech Refresh

The software design is broken into packages of related classes. The table below shows each package to support the new features needed for the mapping technology refresh.

Package Name	Package Description
CHART2.webservices.exportlistenermodule	This package provides the base functionality for the CHART Export Client service. This package is modified to add a new optional data cache.
CHART2.webservices.exportlistenermodule.camera	This package provides the camera specific processing within the CHART Export Client Service. It is modified to maintain an in memory cache of camera objects.
CHART2.webservices.exportlistenermodule.dms	This package provides the DMS specific processing within the CHART Export Client Service. It is modified to maintain an in memory cache of DMS objects.
CHART2.webservices.exportlistenermodule.har	This package provides the HAR specific processing within the CHART Export Client Service. It is modified to maintain an in memory cache of HAR objects.
CHART2.webservices.exportlistenermodule.permit	This package provides the LCP Permit specific processing within the CHART Export Client Service. It is modified to maintain an in memory cache of active and planned permit objects.
CHART2.webservices.exportlistenermodule.restrictions	This package provides the Hauling Restriction specific processing within the CHART Export Client Service. It is modified to maintain an in memory cache of hauling restrictions that is periodically refreshed from the database.
CHART2.webservices.exportlistenermodule.pubsite	This package provides the implementation of the requests that return JSON data to display on the map.

Package Name	Package Description
CHART2.webservices.exportlistenermodule.trafficevent	This package provides the Traffic Event specific processing within the CHART Export Client Service. It is modified to maintain an in memory cache of Traffic Event objects.
CHART2.webservices.exportlistenermodule.tss	This package provides the TSS specific processing within the CHART Export Client Service. It is modified to maintain an in memory cache of TSS objects.
CHART2.DMSUtility.multi	This package provides MULTI parsing capabilities. A new formatter is added that can create an HTML representation of a DMS message from MULTI.

3.6 Assumptions and Constraints

3.6.1 CHARTWeb Map Refresh

1. Assumptions and Constraint: The EORS V2 data (Snow Emergency Plans, Hauling Restrictions, IPS) will only update on the map as frequently as the existing database copy job makes the data available in the CHARTWeb mapping database.

3.6.2 PRs

1. Assumptions or Constraints: None.

4 Human Machine Interface

4.1 CHARTWeb

This section describes the CHARTWeb Map Refresh User Interface

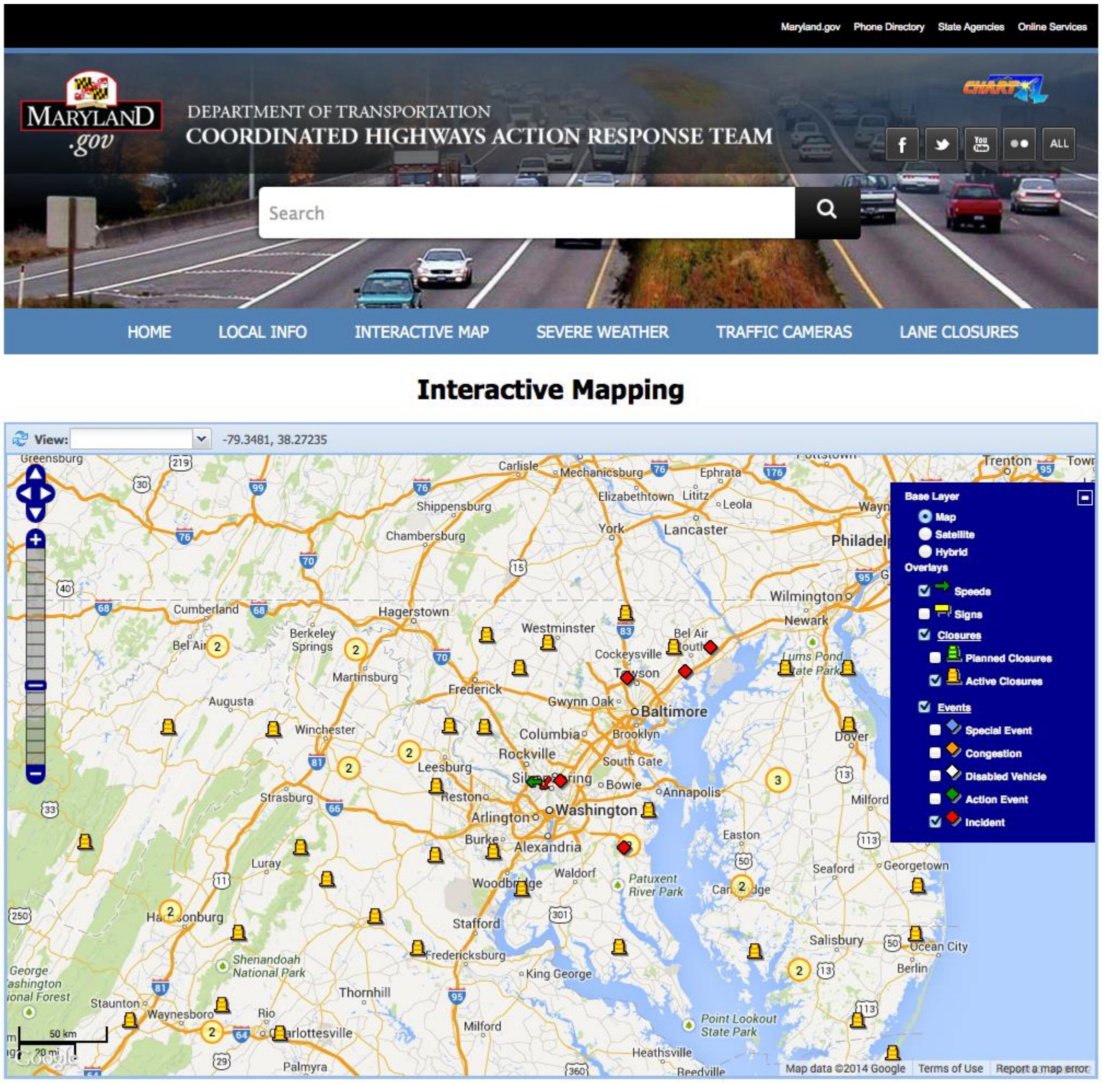
4.1.1 Background Information

The CHARTWeb Map Refresh application will completely replace the existing public web site map. The goals of the refresh are to make the map more user friendly and more efficient. The page will use a Google Map as the default base map.

4.1.2 CHARTWeb Map Refresh

4.1.2.1 Default Map View

The default map view replaces the existing tabs “Traffic, Road Work, Hauling Restrictions, Roadway Weather, Video Camera, Info Devices” with a single Google map as shown below. The user can utilize the “layer switcher” control at the right side of the screen to control which map layers are visible at any time. By default the Speeds, Active Closures, and Incidents layers will be the only visible layers. The user can also use the layer switcher to select the base map view that they would like to view. The initial options will be Map (Google Streets), Satellite (Google Satellite), and Hybrid (Google Hybrid). Other base maps can be added in the future as desired.

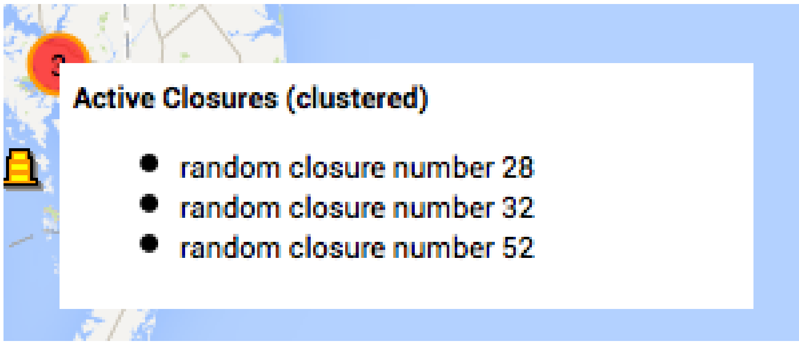


Base Layer
- ● Map
- ○ Satellite
- ○ Hybrid

Overlays
- ☑ Speeds
- ☐ Signs
- ☑ Closures
  - ☐ Planned Closures
  - ☑ Active Closures
- ☑ Events
  - ☐ Special Event
  - ☐ Congestion
  - ☐ Disabled Vehicle
  - ☐ Action Event
  - ☑ Incident

Figure 4-1. The Updated Default Map View.

The map will use the "clustering" strategy for layers that have a large number of data items to display that are too close to one another. In the image above the yellow circles with orange outlines represent areas where there are more than 1 active permit in close proximity to one another. The number in the circle shows the user how many permits the circle represents. The user may hover over the cluster icon in order to see a tooltip that provides the description of each permit in the cluster as shown below. Notice also that the cluster changes color to show that it is the selected cluster.

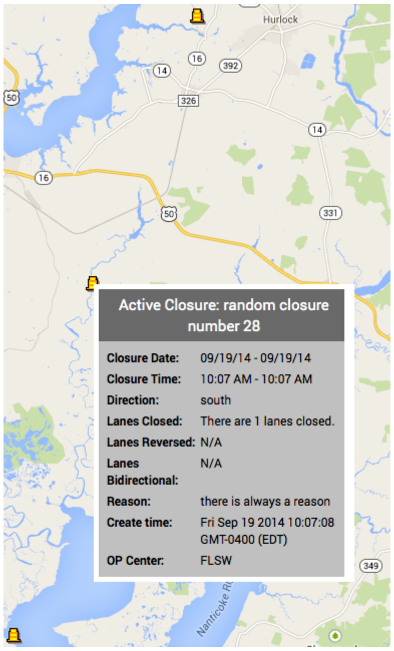


Active Closures (clustered)

- random closure number 28
- random closure number 32
- random closure number 52

Figure 4-2. Cluster Tool Tip.

Clicking on the cluster will zoom the map to the area of the clustered objects to allow the user to see the individual items that are contained within the cluster. The image below shows the map after clicking on the cluster shown above. Notice that the user can now hover over any of the active closure icons in order to get more information about the closure in question. This is always the case. Each non-clustered icon, on any layer, can be hovered over to show details about the data item.



Active Closure: random closure number 28

Closure Date:	09/19/14 - 09/19/14
Closure Time:	10:07 AM - 10:07 AM
Direction:	south
Lanes Closed:	There are 1 lanes closed.
Lanes Reversed:	N/A
Lanes Bidirectional:	N/A
Reason:	there is always a reason
Create time:	Fri Sep 19 2014 10:07:08 GMT-0400 (EDT)
OP Center:	FLSW

Figure 4-3. Active Closure Tool Tip.

By default the map will initially show an overview of the state of Maryland. The user can select from a set of pre-defined map views by selecting one from the select box labeled “View” on the toolbar directly above the map as shown below. The list of views will be populated with the same list that is currently available on the public web site. Clicking on an entry in the list will pan and zoom the map so that the selected area is in the map viewport.

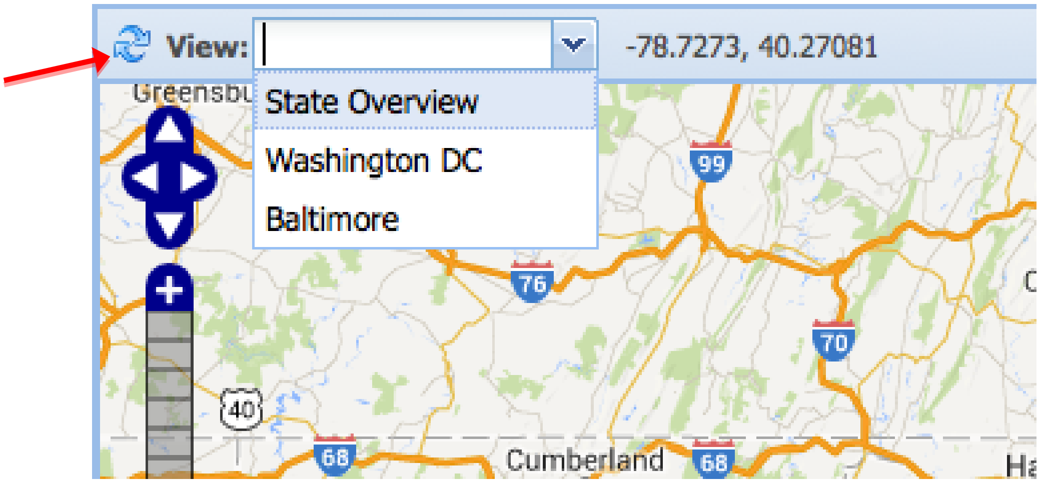


Figure 4-4. Pre-defined Map View Selection.

The map will always update the data on all layers automatically based on a timer that fires on an interval that can be configured by an administrator (default is 1 minute). However, the user can refresh their map data at any point in time by clicking the refresh button on the toolbar to the left of the view selector (pointed to by the red arrow in the image above).

By default the map will open within that context of the CHART Public Web Site as shown in Figure 4-1. However, the user will have the option of launching their map view into a separate browser window or tab that is dedicated entirely to the map display as shown above. If the user re-sizes their browser window this map will always resize to fill the entirety of the browser display with the map. This can be seen in figures 4-5 and 4.6 below. With the exception of the responsive page design, this map functions exactly the same as the map that is embedded within the web site.

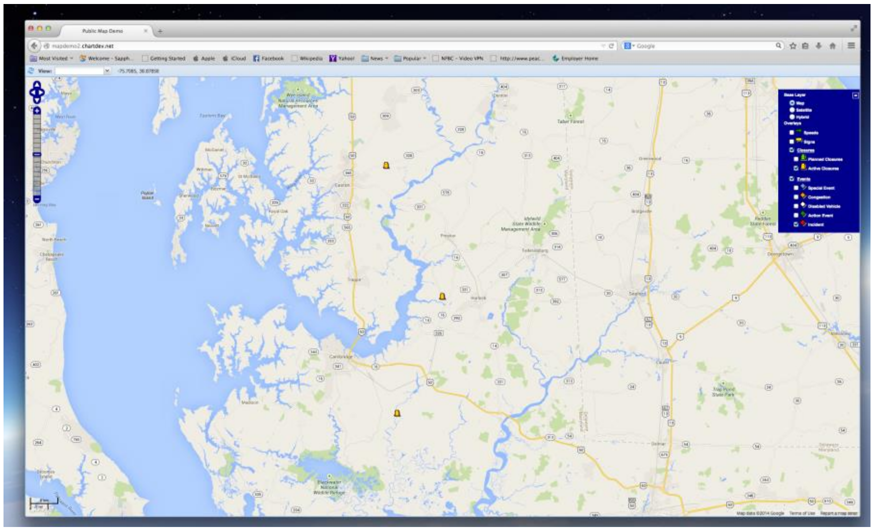


Figure 4-5. Full Browser Map View – Large Browser Window.

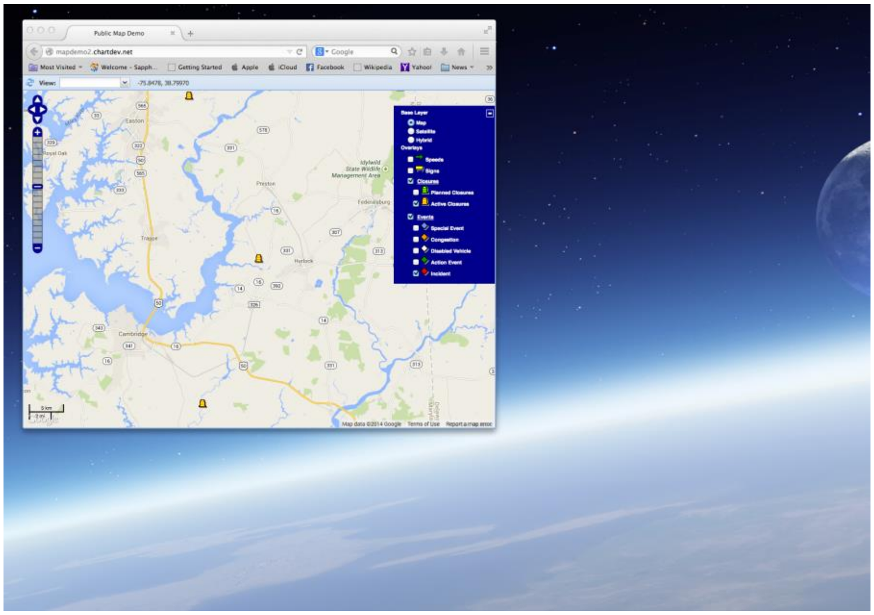


Figure 4-6. Full Browser Map View – Smaller Browser Window.

5 Deprecated Functionalities

5.1 CHARTweb

The existing map and all related code will be completely removed from the site.

6 Acronyms/Glossary

LCP	Lane Closure Permits application used to manage SHA and MDTA lane closure permitting.
Home Page	The main page of the LCP application, always open if the user is logged in.
Permit	An LCP permit, providing information about roadwork that is permitted to take place.
RAM	Roadway Approval Manager
DPT	District Approval By Permit Type
WML	Workflow Manager Log
GIS	**Geographic Information System** (GIS) is any system that captures, stores, analyzes, manages, and presents data that are linked to location
Intranet Map	The CHART Mapping application that is not integrated into the CHART user interface.
REST	Representational State Transfer - a web services architecture style used in CHART that leverages web technologies such as http and XML

7 Use Case Diagrams

7.1 Feature (Use Case Diagram)

This diagram shows the possible uses of the public web site map.

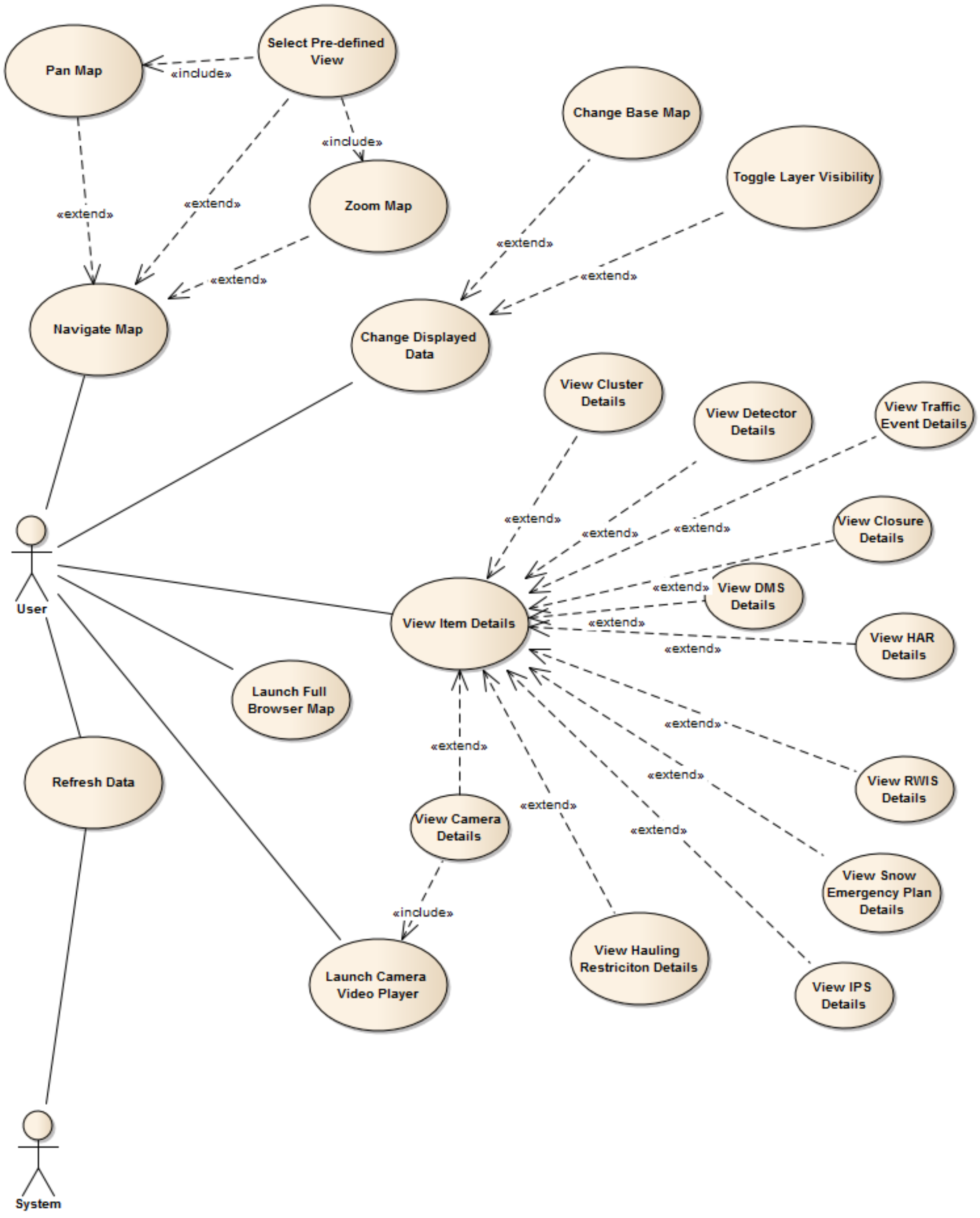


Figure 7-1 Public Site Web Map Use Cases

7.1.1 Navigate Map (Use Case)

A user can navigate the map by panning, zooming or selecting a pre-defined map view.

7.1.2 Pan Map (Use Case)

A user can pan the map using the pan control or by clicking on the map and dragging it.

7.1.3 Select Pre-defined View (Use Case)

A user can pan and zoom the map to a pre-defined area by selecting the area name from a list of available views.

7.1.4 Zoom Map

A user can zoom the map by double clicking on a point where there are no icons, clicking on the base map and dragging while holding the control key (rubber band zoom), or by using the mouse wheel (if their mouse has a wheel).

7.1.5 Change Displayed Data

A user can change the data that is currently displayed on the map. This can be done by changing the base map type, or by changing the visibility of one of the data layers. Available layers include: Detectors (Speed), Traffic Events (a layer per type), LCP Permits (Closures), DMS signs, HARs, Cameras, RWIS detectors, Snow Emergency Plans, IPS data and Hauling Restrictions.

7.1.6 Change Base Map

The user can use the layer selector to change the base map from a list of available base map types.

7.1.7 Toggle Layer Visibility

The user can toggle the visibility of any layer independently, or can toggle groups of related layers.

7.1.8 View Item Details

A user can hover over an icon in order to see more details about the object.

7.1.9 View Cluster Details

A user can hover over a cluster icon in order to view the list of objects that the cluster represents. A name is shown for each object.

7.1.10 View Detector Details

A user can hover over a TSS arrow to see details about the speed at that location. If the TSS is not CHART owned it will show only a speed range at the location instead of a specific speed.

7.1.11 View Traffic Event Details

A user may view the details of a traffic event by hovering over a traffic event icon on the map.

7.1.12 View Closure Details

A user may view the details of an LCP Permit (closure) by hovering over a closure icon on the map.

7.1.13 View DMS Details

A user may view the details of a DMS by hovering over a DMS icon on the map. The details panel that is displayed will include a graphical representation of the current message.

7.1.14 View HAR Details

A user may view the details of a HAR by hovering over a HAR icon on the map.

7.1.15 View Camera Details

A user may view the details of a Camera by hovering over a Camera icon on the map.

7.1.16 Launch Camera Video Player

A user may launch a video player for a camera that has an available video stream. If the camera is currently blocked to public this action will not be available.

7.1.17 View RWIS Details

A user may view the details of an RWIS by hovering over an RWIS icon on the map.

7.1.18 View Snow Emergency Plan Details

A user may view the details of a Snow Emergency Plan by hovering over a Snow Emergency Plan icon on the map.

7.1.19 View IPS Details

A user may view the details of an IPS by hovering over an IPS icon on the map.

7.1.20 View Hauling Restriction Details

A user may view the details of a hauling restriction by hovering over a restriction icon on the map.

7.1.21 Launch Full Browser Map

A user can launch a full browser map to a new window or tab. The map that is launched will always size to fill the entire browser window/tab it is in.

7.1.22 Refresh Data

The system will automaticallly refresh all data on a pre-configured interval. A user can request a data refresh at any time.

8 Package Designs

For convenient viewing, new and modified package designs are included in a separate document for viewing with a browser. Open the file `index.htm`. See figure 8-1 below to see the entry point in the design document for the class and sequence diagrams mentioned below. Chartweb-WO47-pubmap -> classes -> CHART2 is the top level package.

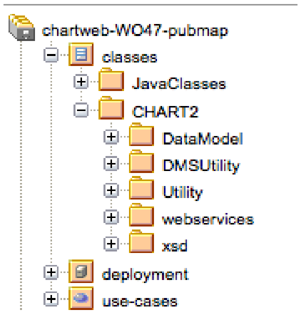


Figure 2: How to find sequence and class diagrams

8.1 ExportListenerModule Changes

The figure below shows where the ExportListenerModule class diagram is located within the design.

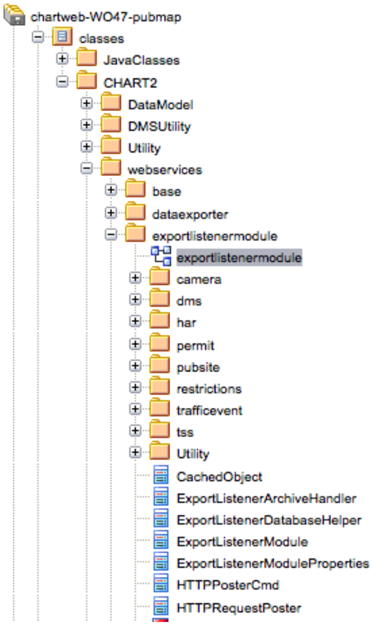
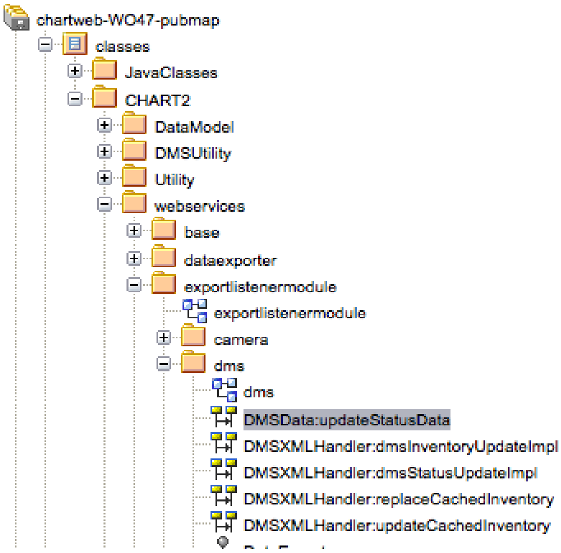


Figure 3: Location of ExportListenerModule class diagram

8.2 ExportListenerModule.dms Package Changes

The figure below shows where the ExportListenerModule.dms sequence diagrams are located within the design.

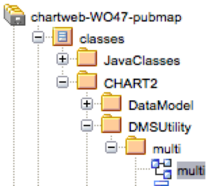


- chartweb-WO47-pubmap
  - classes
    - JavaClasses
    - CHART2
      - DataModel
      - DMSUtility
      - Utility
      - webservices
        - base
        - dataexporter
        - exportlistenermodule
          - exportlistenermodule
          - camera
          - dms
            - dms
            - DMSData:updateStatusData
            - DMSXMLHandler:dmsInventoryUpdateImpl
            - DMSXMLHandler:dmsStatusUpdateImpl
            - DMSXMLHandler:replaceCachedInventory
            - DMSXMLHandler:updateCachedInventory

Figure 4: Location of ExportListenerModule.dms sequence diagram

8.3 DMSUtility.multi Package Changes

The figure below shows where the DMSUtility.multi class diagram is located within the design.

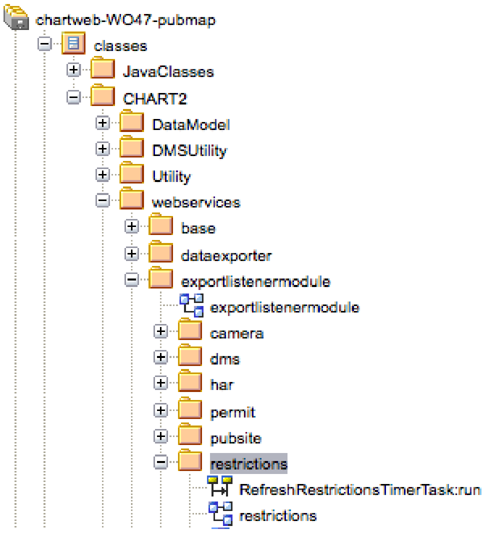


- chartweb-WO47-pubmap
  - classes
    - JavaClasses
    - CHART2
      - DataModel
      - DMSUtility
        - multi
          - multi

Figure 5: Location of DMSUtility.multi class diagram

8.4 ExportListenerModule.restrictions Package Changes

The figure below shows where the ExportListenerModule.restrictions class and sequence diagrams are located within the design.

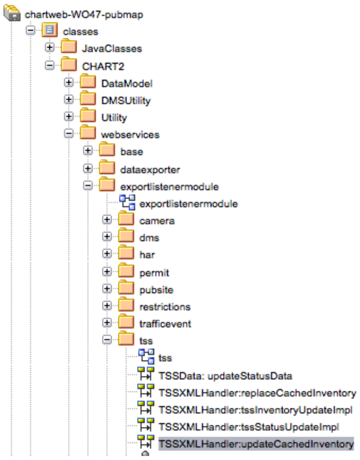


```
chartweb-WO47-pubmap
  classes
    JavaClasses
    CHART2
      DataModel
      DMSUtility
      Utility
      webservices
        base
        dataexporter
        exportlistenermodule
          exportlistenermodule
          camera
          dms
          har
          permit
          pubsite
          restrictions
            RefreshRestrictionsTimerTask:run
            restrictions
```

Figure 6: Location of ExportListenerModule.restrictions diagrams

8.5 ExportListenerModule.tss Package Changes

The figure below shows where the ExportListenerModule.tss class and sequence diagrams are located within the design.

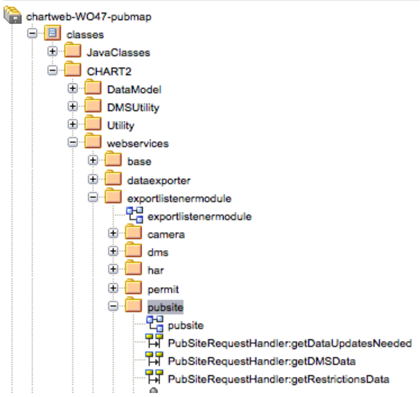


```
chartweb-WO47-pubmap
  classes
    JavaClasses
    CHART2
      DataModel
      DMSUtility
      Utility
      webservices
        base
        dataexporter
        exportlistenermodule
          exportlistenermodule
          camera
          dms
          har
          permit
          pubsite
          restrictions
          trafficevent
          tss
            tss
            TSSData: updateStatusData
            TSSXMLHandler:replaceCachedInventory
            TSSXMLHandler:tssInventoryUpdateImpl
            TSSXMLHandler:tssStatusUpdateImpl
            TSSXMLHandler:updateCachedInventory
```

Figure 7: Location of ExportListenerModule.tss diagrams

8.6 ExportListenerModule.pubsite Package Changes

The figure below shows where the ExportListenerModule.pubsite class and sequence diagrams are located within the design.



```
chartweb-WO47-pubmap
  classes
    JavaClasses
    CHART2
      DataModel
      DMSUtility
      Utility
      webservices
        base
        dataexporter
        exportlistenermodule
          exportlistenermodule
          camera
          dms
          har
          permit
          pubsite
            pubsite
            PubSiteRequestHandler:getDataUpdatesNeeded
            PubSiteRequestHandler:getDMSData
            PubSiteRequestHandler:getRestrictionsData
```

Figure 8: Location of ExportListenerModule.pubsite diagrams